

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

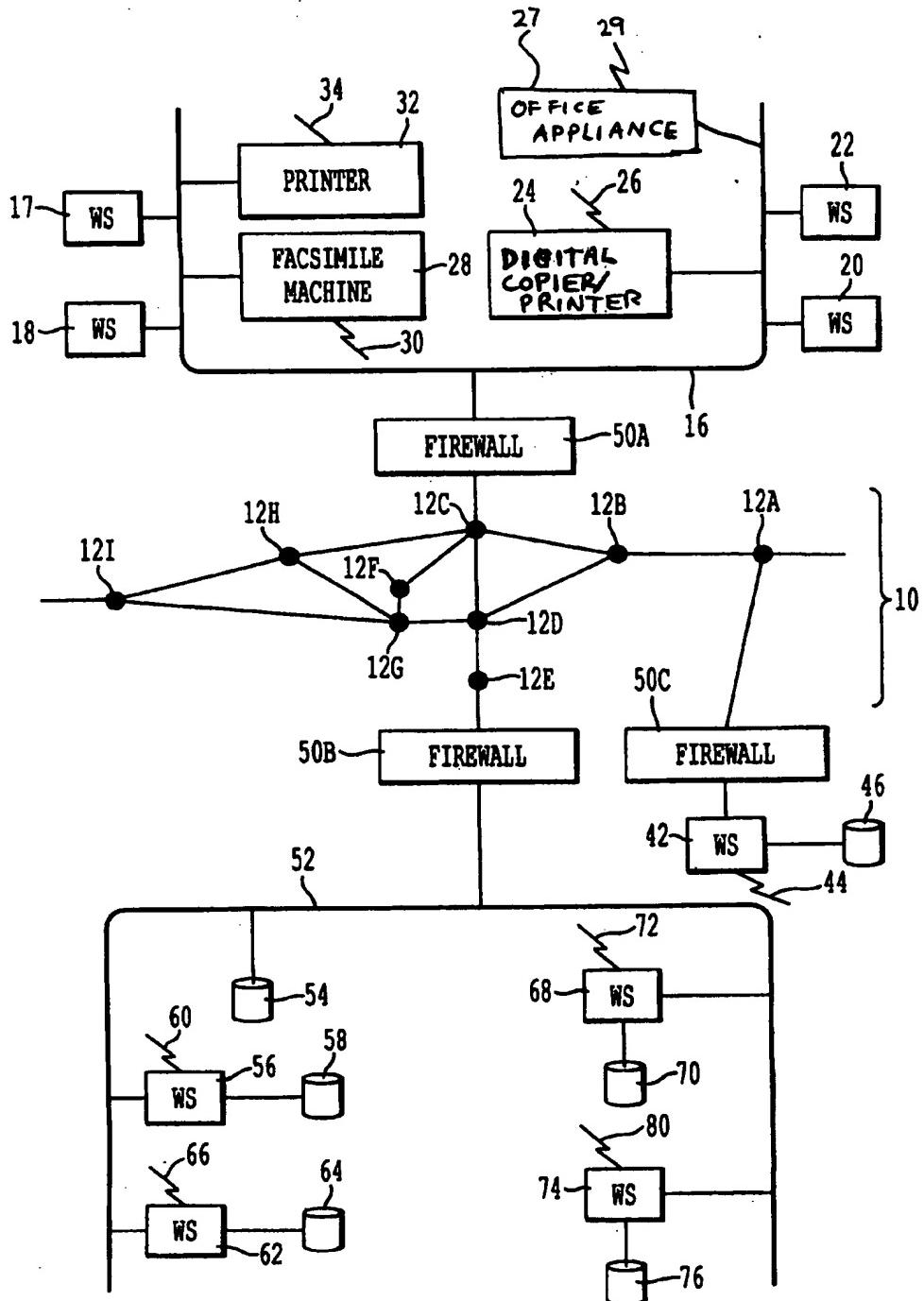


FIG. 1

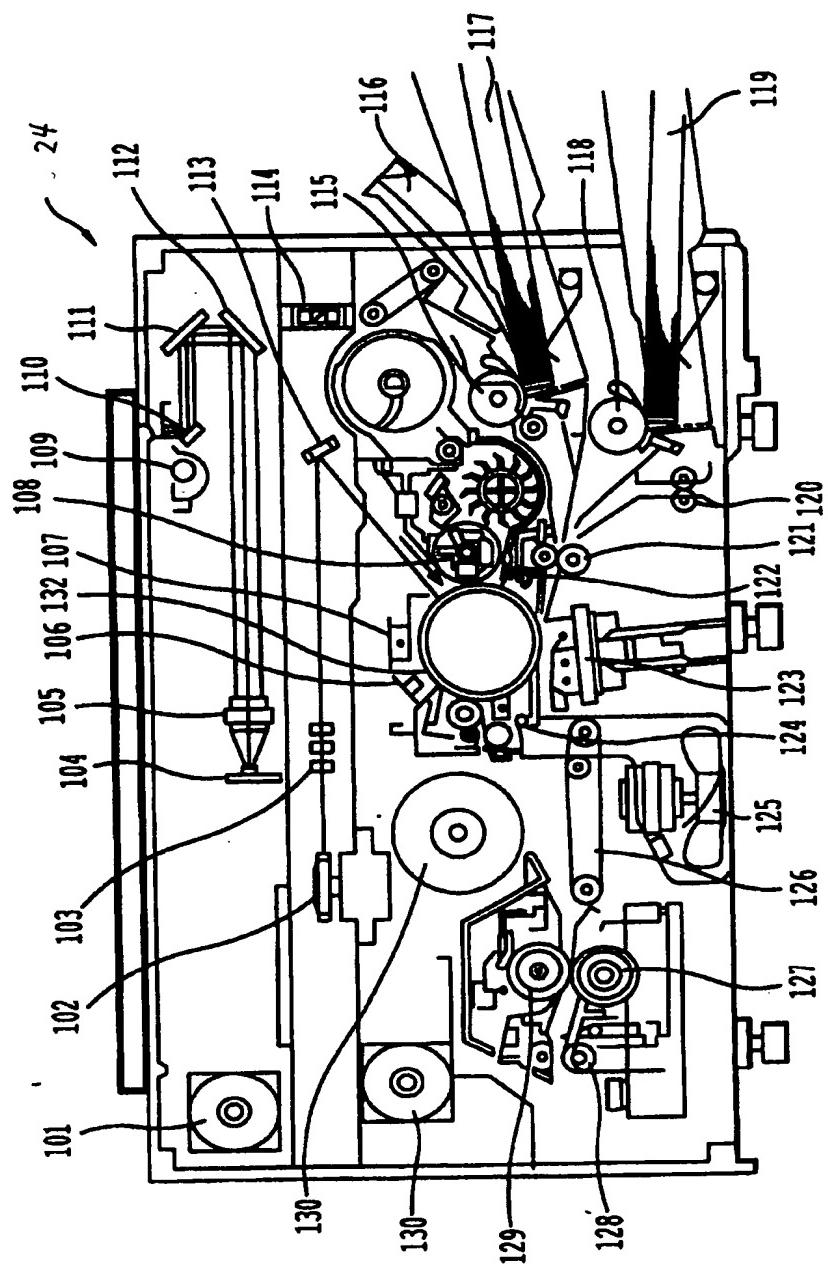


FIG. 2

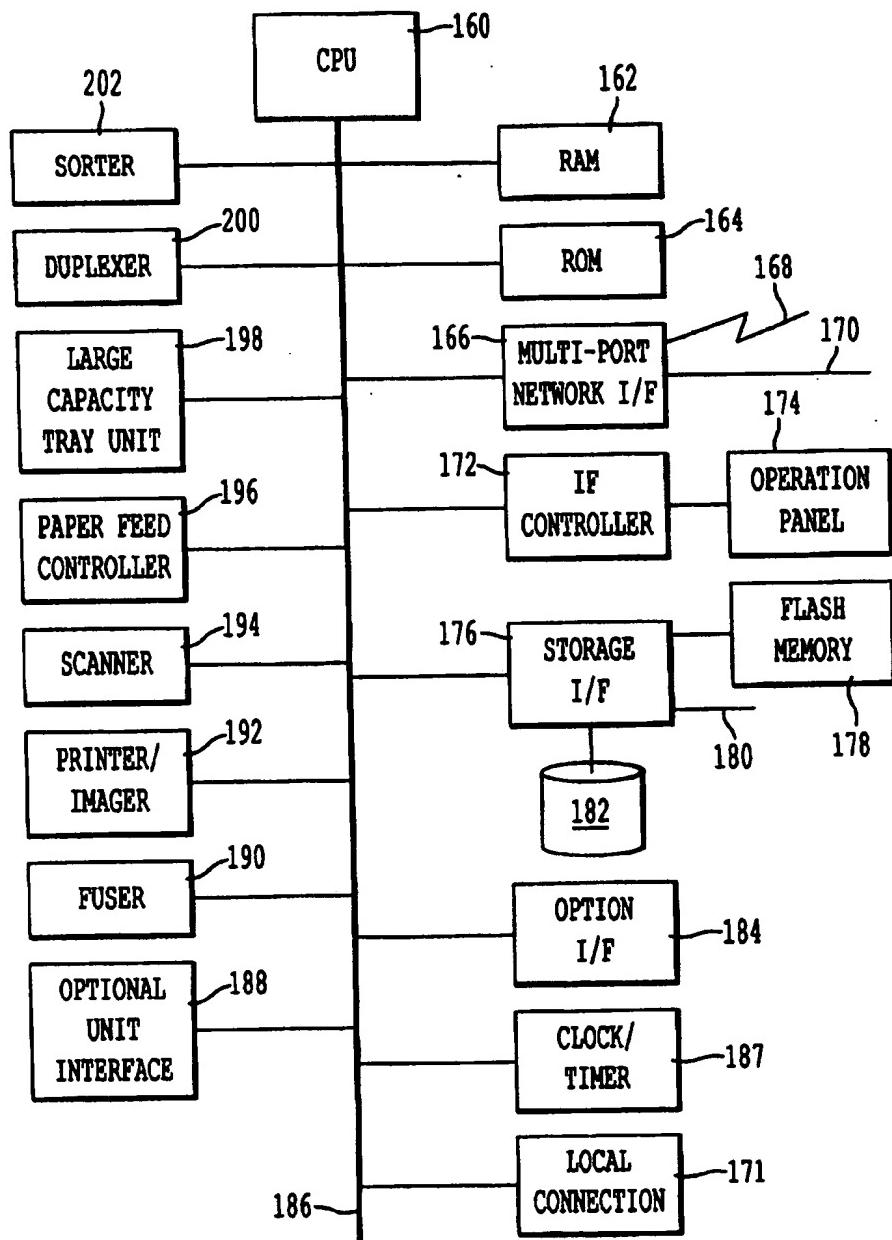


FIG. 3

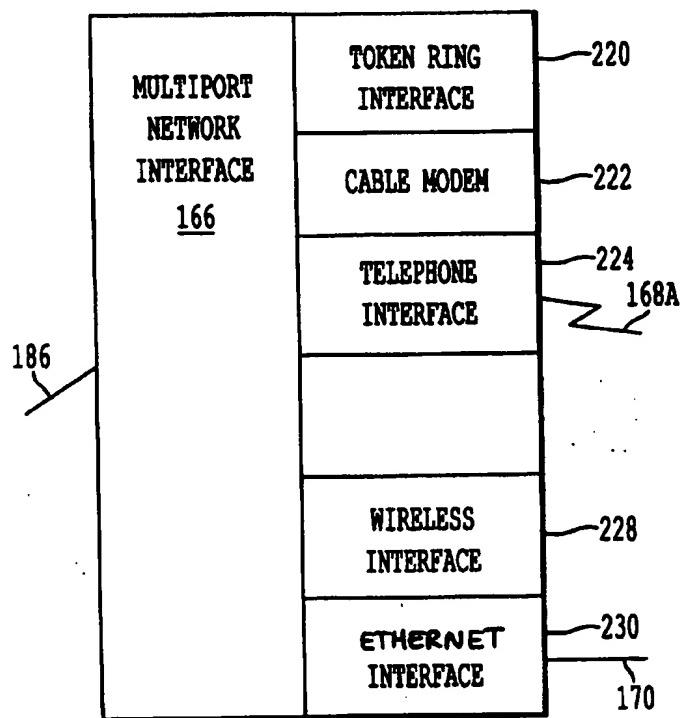


FIG. 4

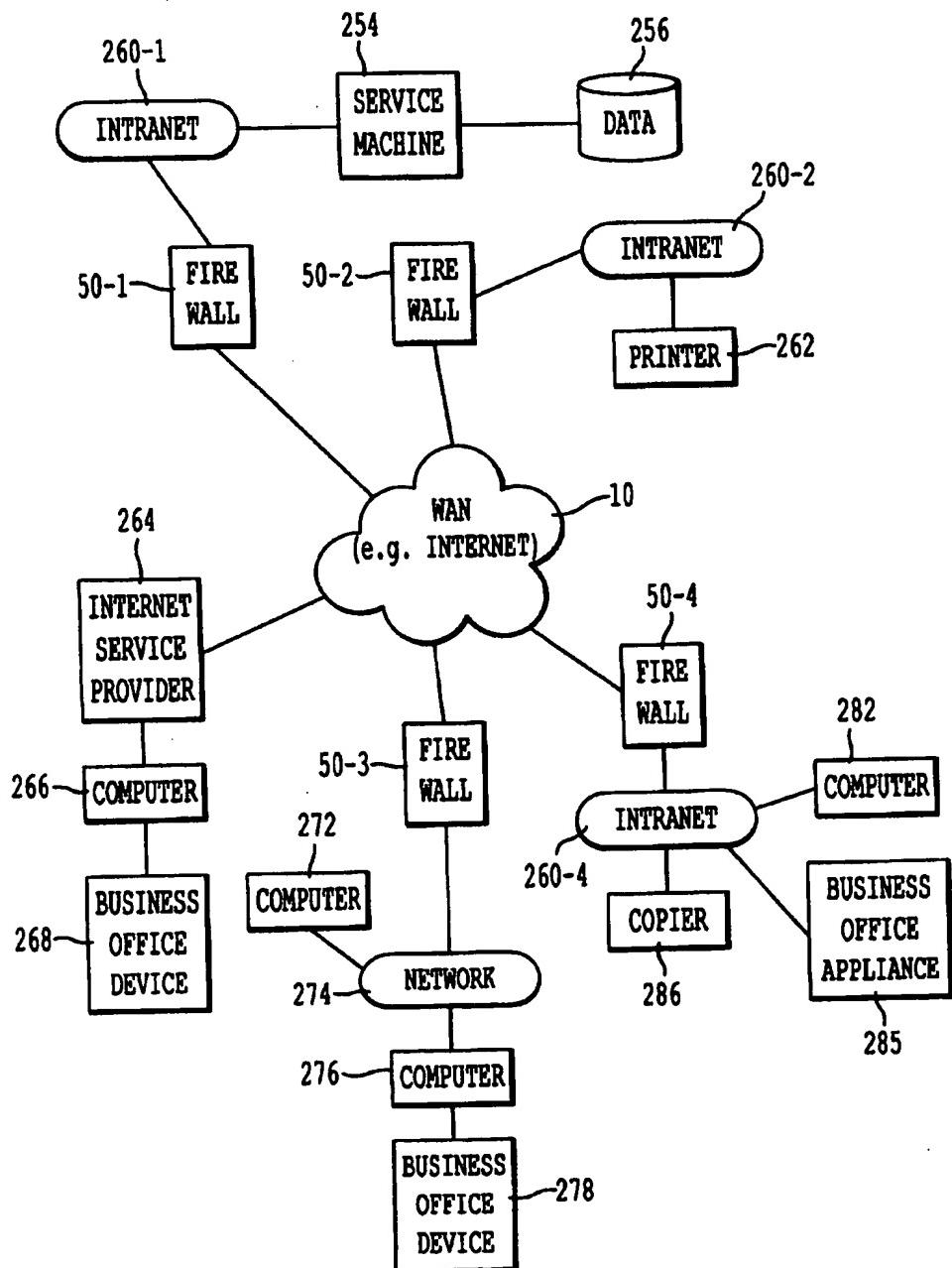


FIG. 5

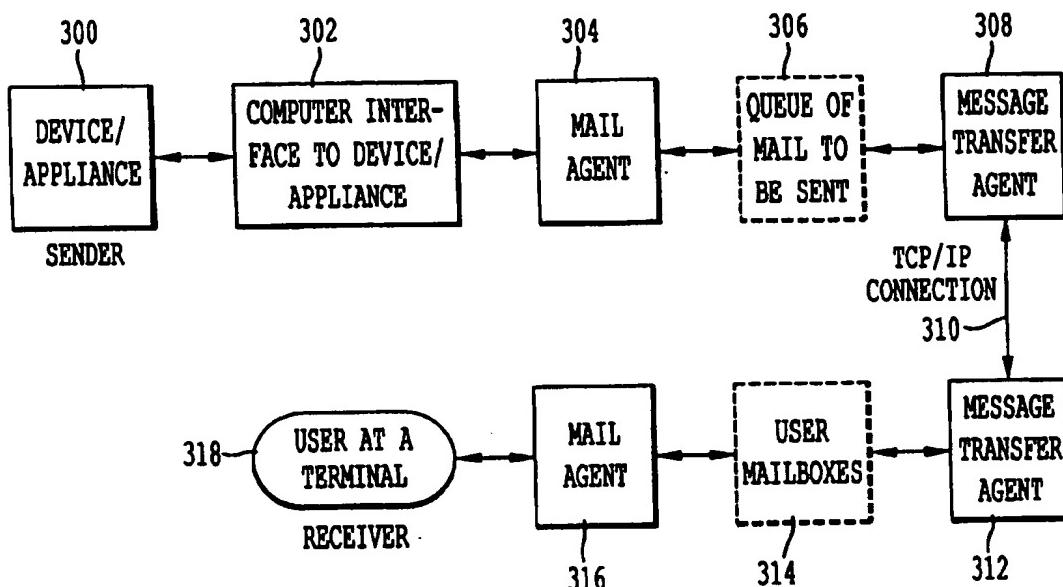


FIG. 6A

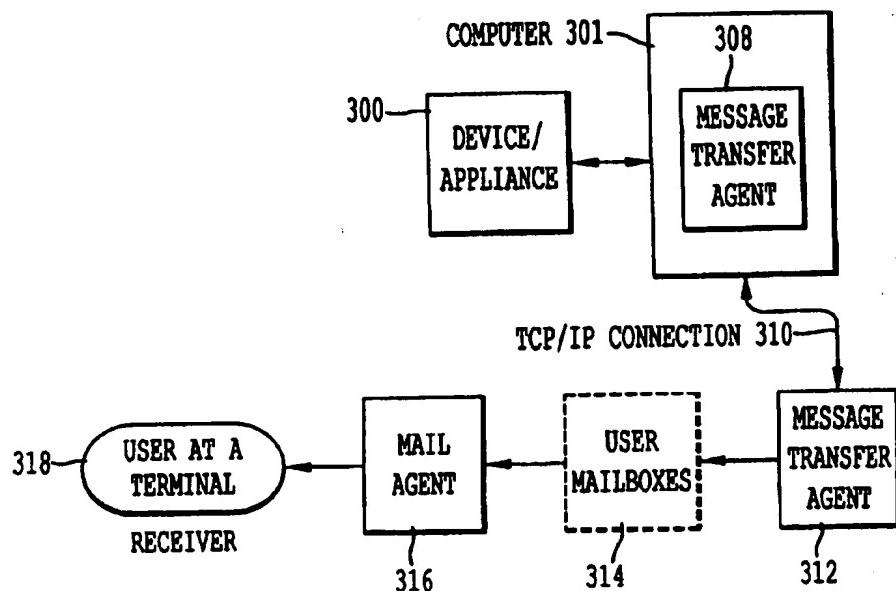


FIG. 6B

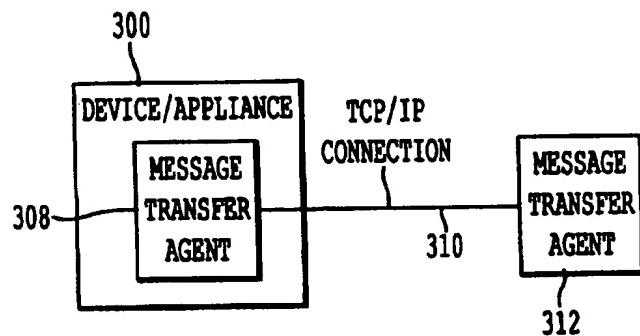


FIG. 6C

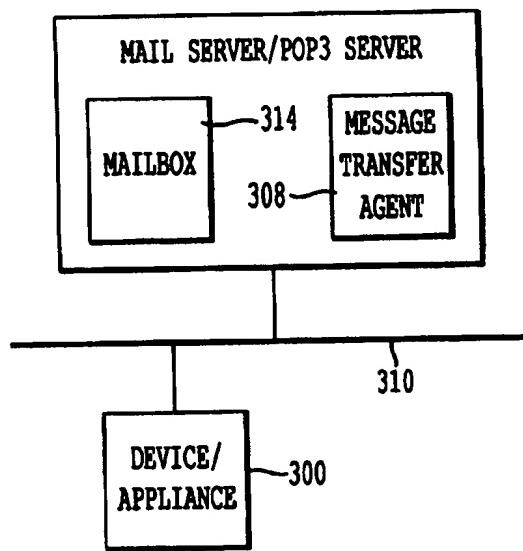


FIG. 6D

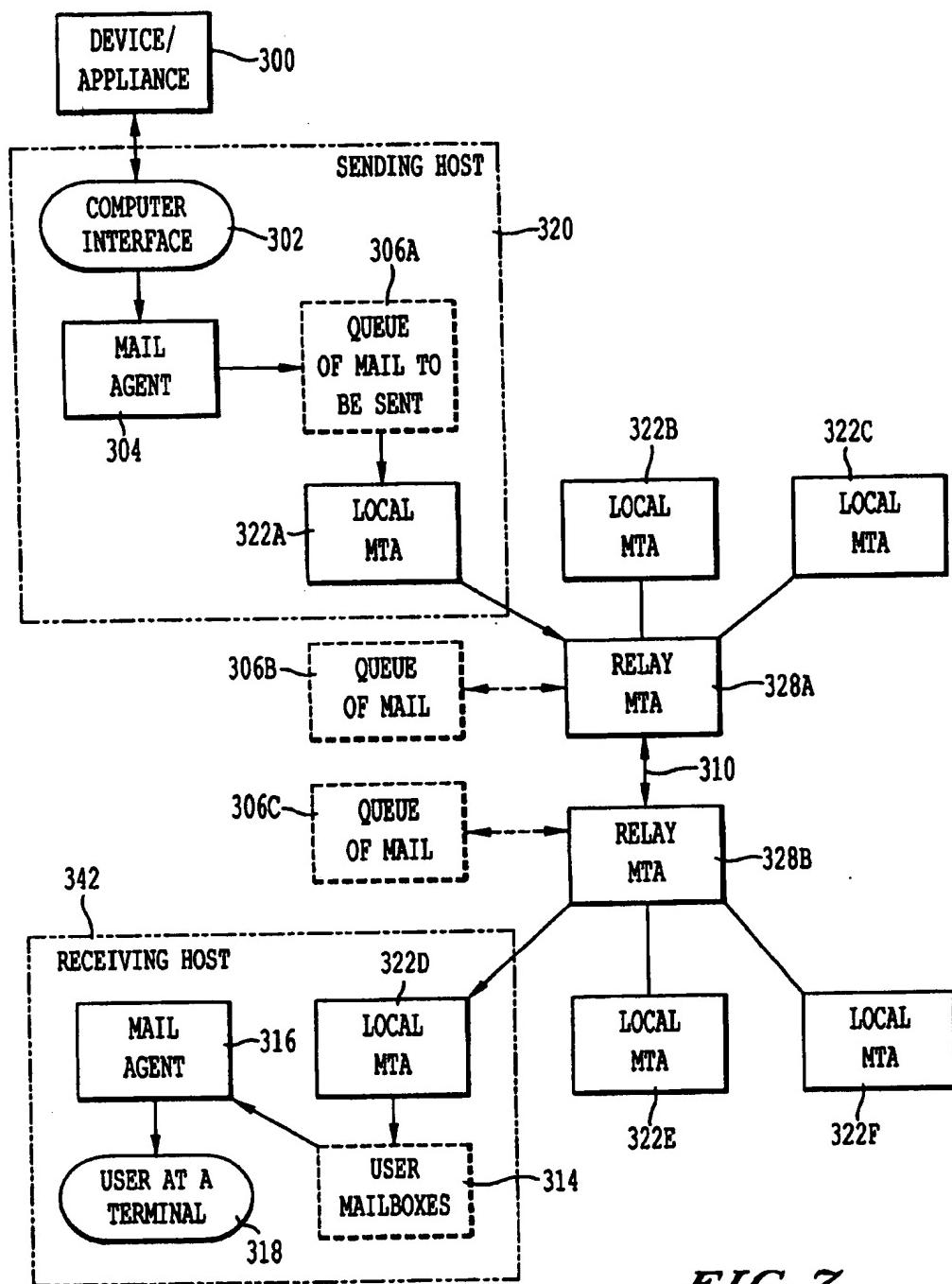


FIG. 7

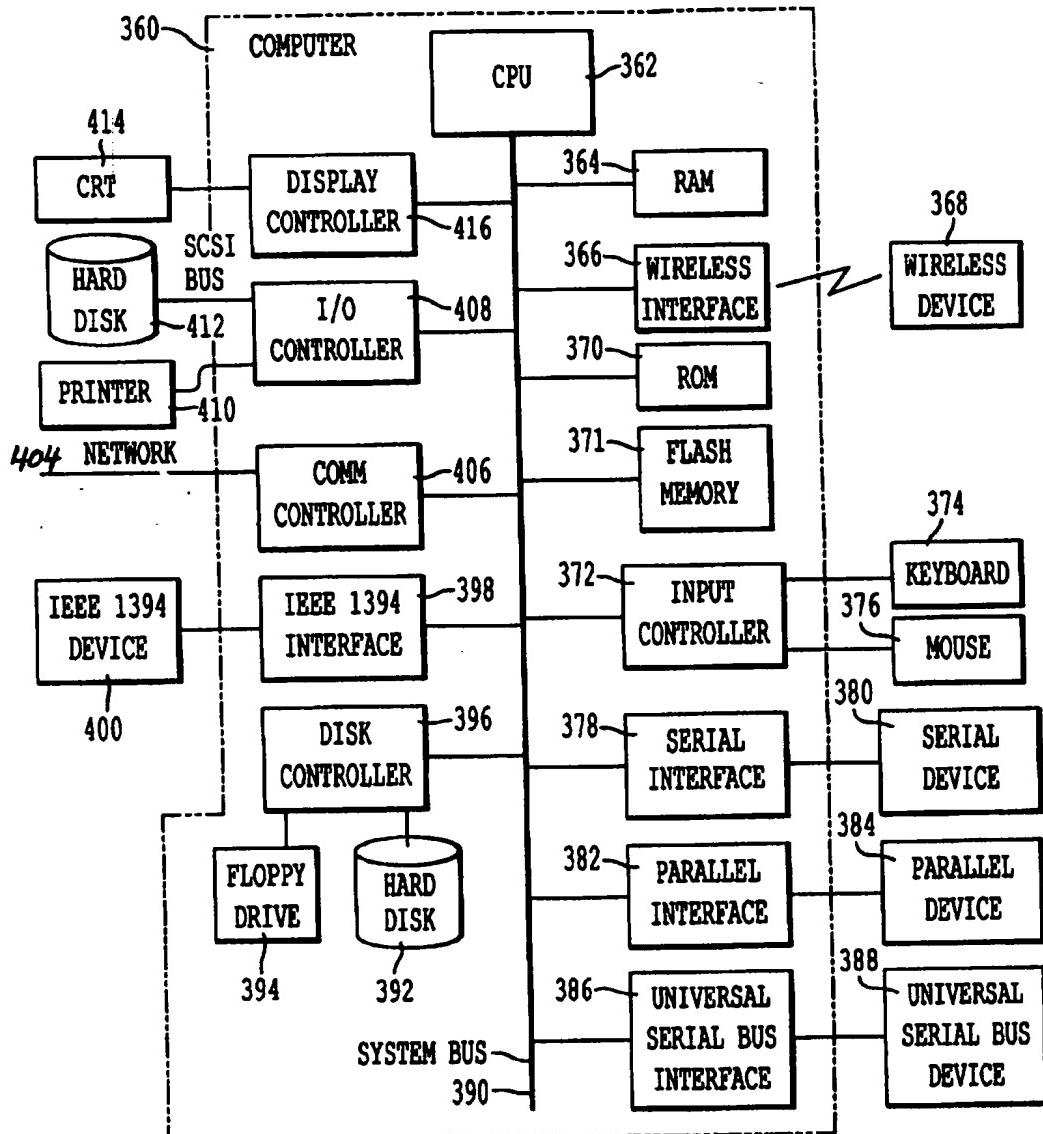


FIG. 8

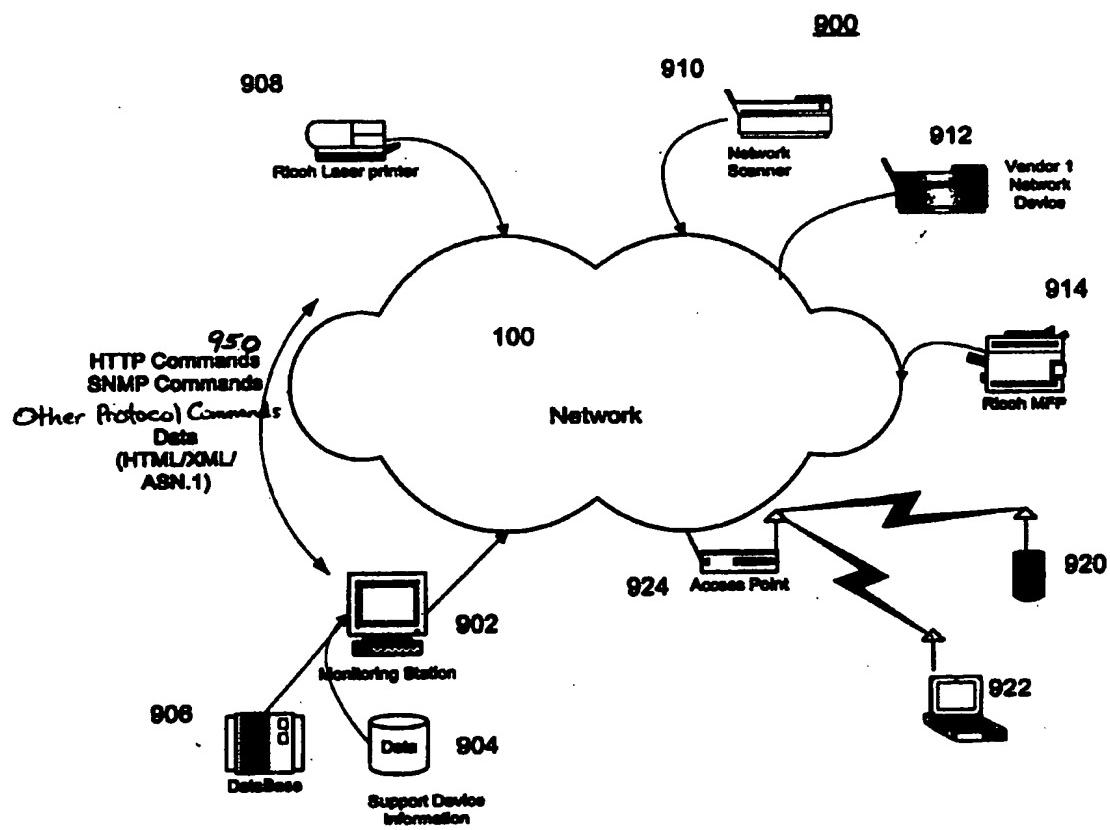


Figure 9

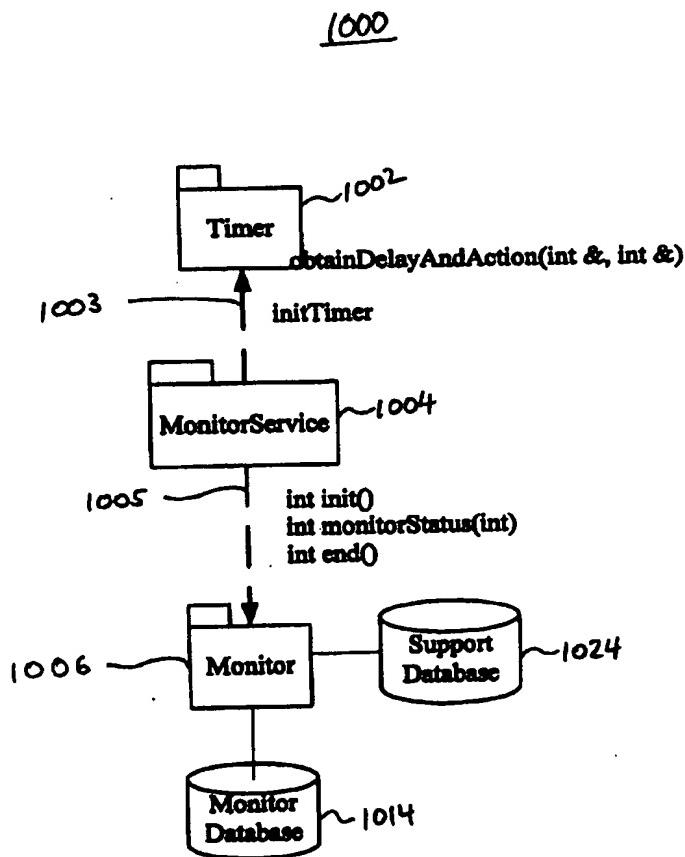


FIG. 10

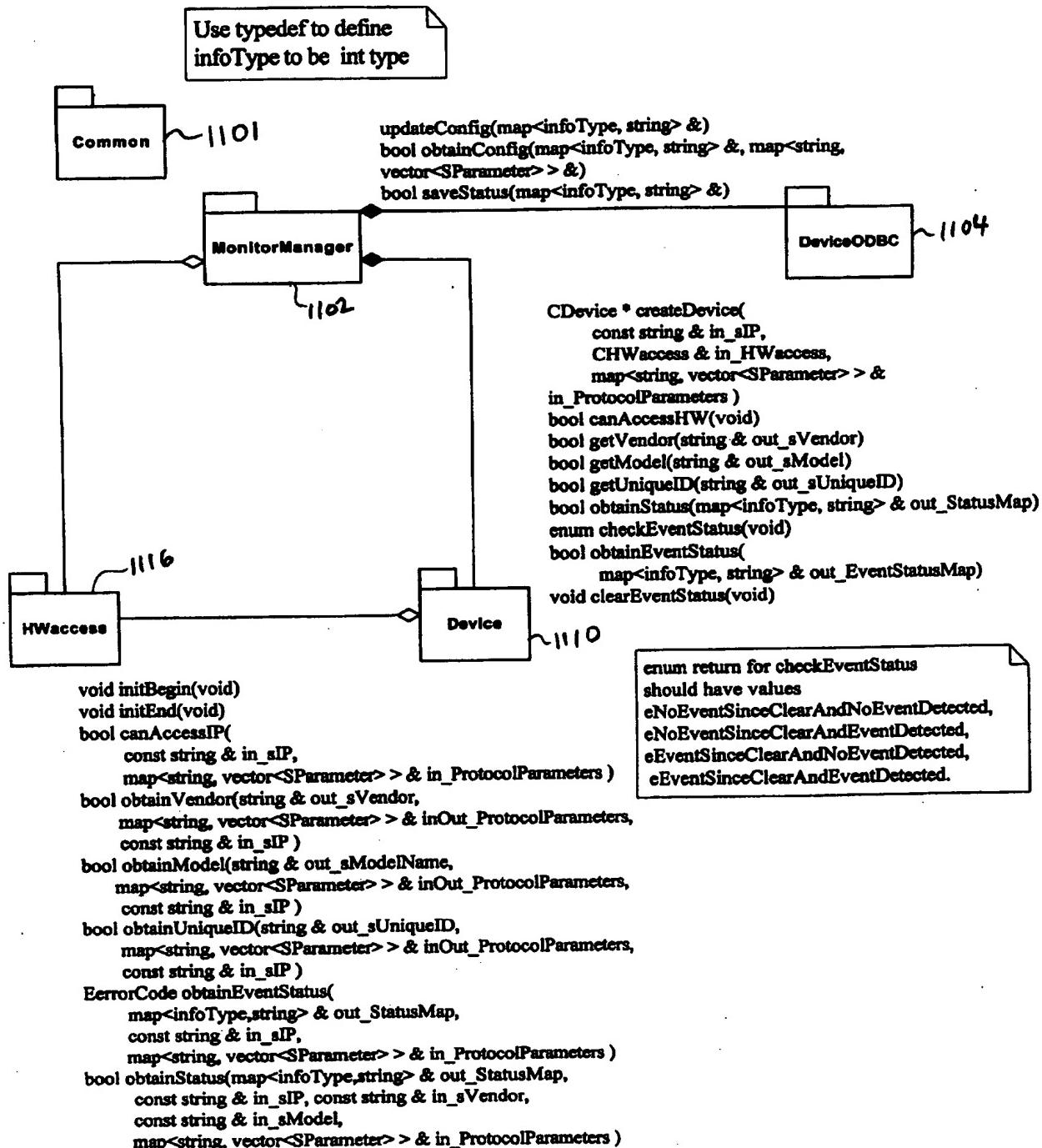


FIG. 11

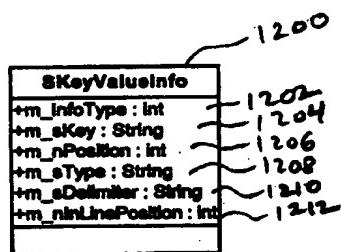


FIG. 12

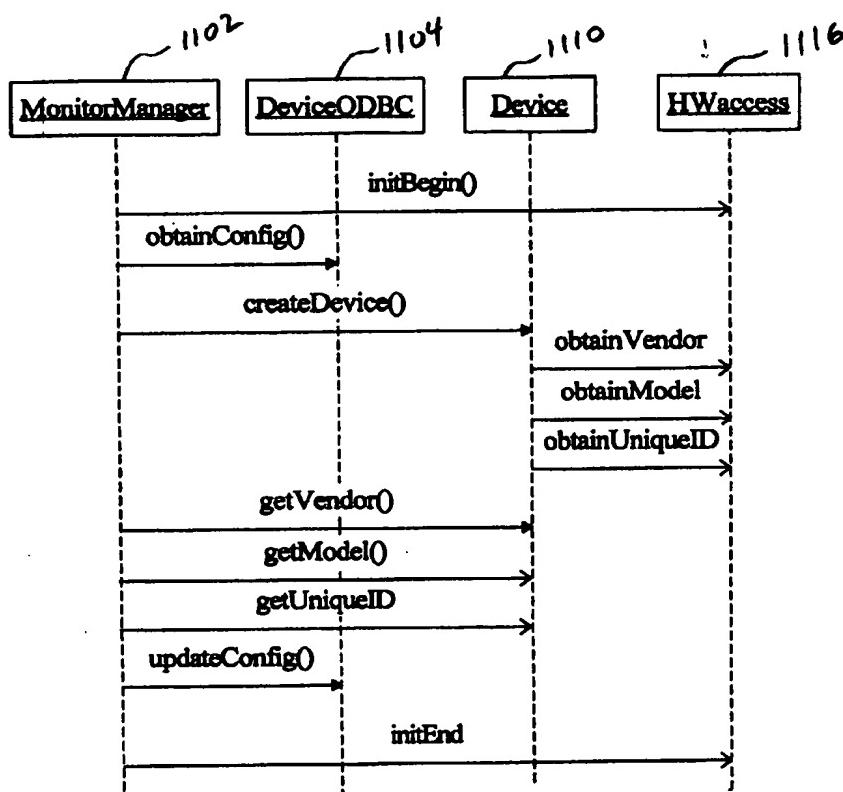


FIG. 13

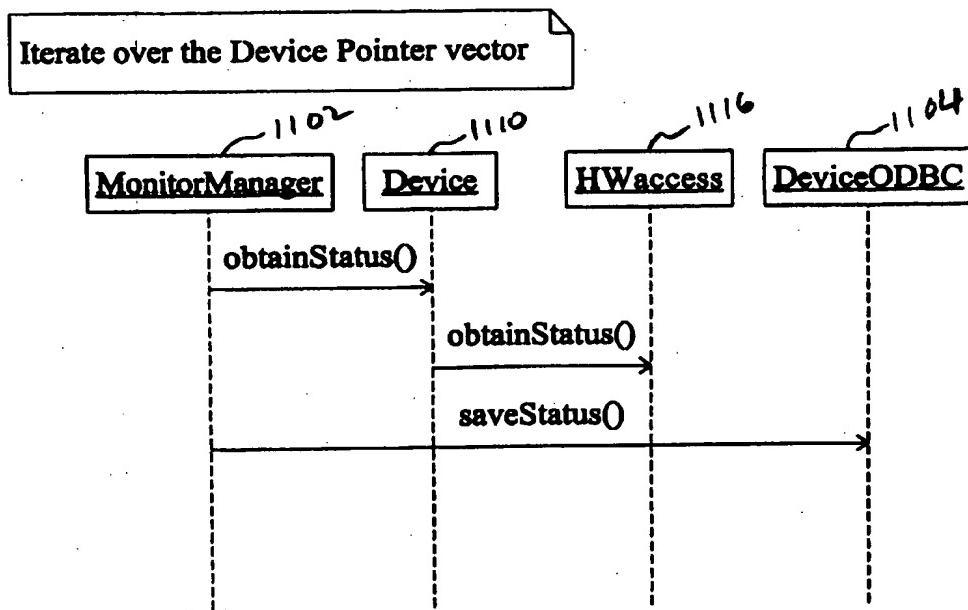
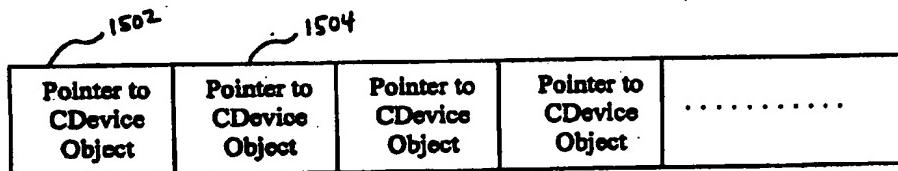


FIG. 14

1500



`std::vector <pointer of CDevice >`

FIG. 15

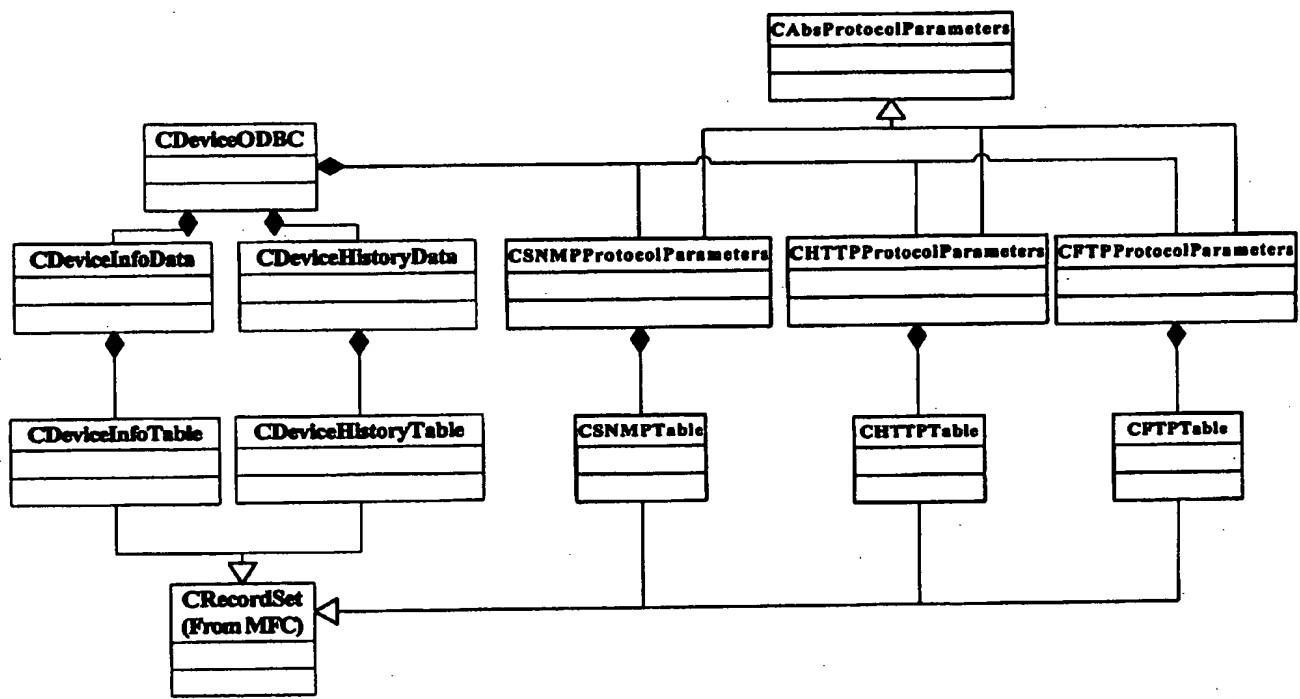


FIG. 16

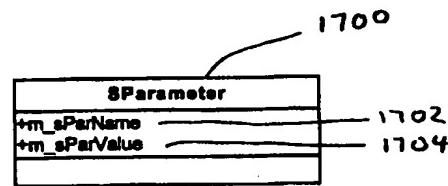


FIG. 17

A diagram showing a table with two columns. The left column is labeled "Protocol (Key)" and the right column is labeled "Vector of SParameter (Value)". There are three rows in the table. The first row contains the value "SNMP" in the "Protocol (Key)" column and is associated with the label "1802" on its left. The second row contains the value "HTTP" in the "Protocol (Key)" column and is associated with the label "1804" on its left. The third row contains the value "FTP" in the "Protocol (Key)" column and is associated with the label "1806" on its left. The "Vector of SParameter (Value)" column for each row is associated with a label on its right: "1808" for the first row, "1810" for the second row, and "1812" for the third row. A curved arrow points from the label "1800" to the top of the table.

Protocol (Key)	Vector of SParameter (Value)
"SNMP"	1808
"HTTP"	1810
"FTP"	1812

FIG. 18

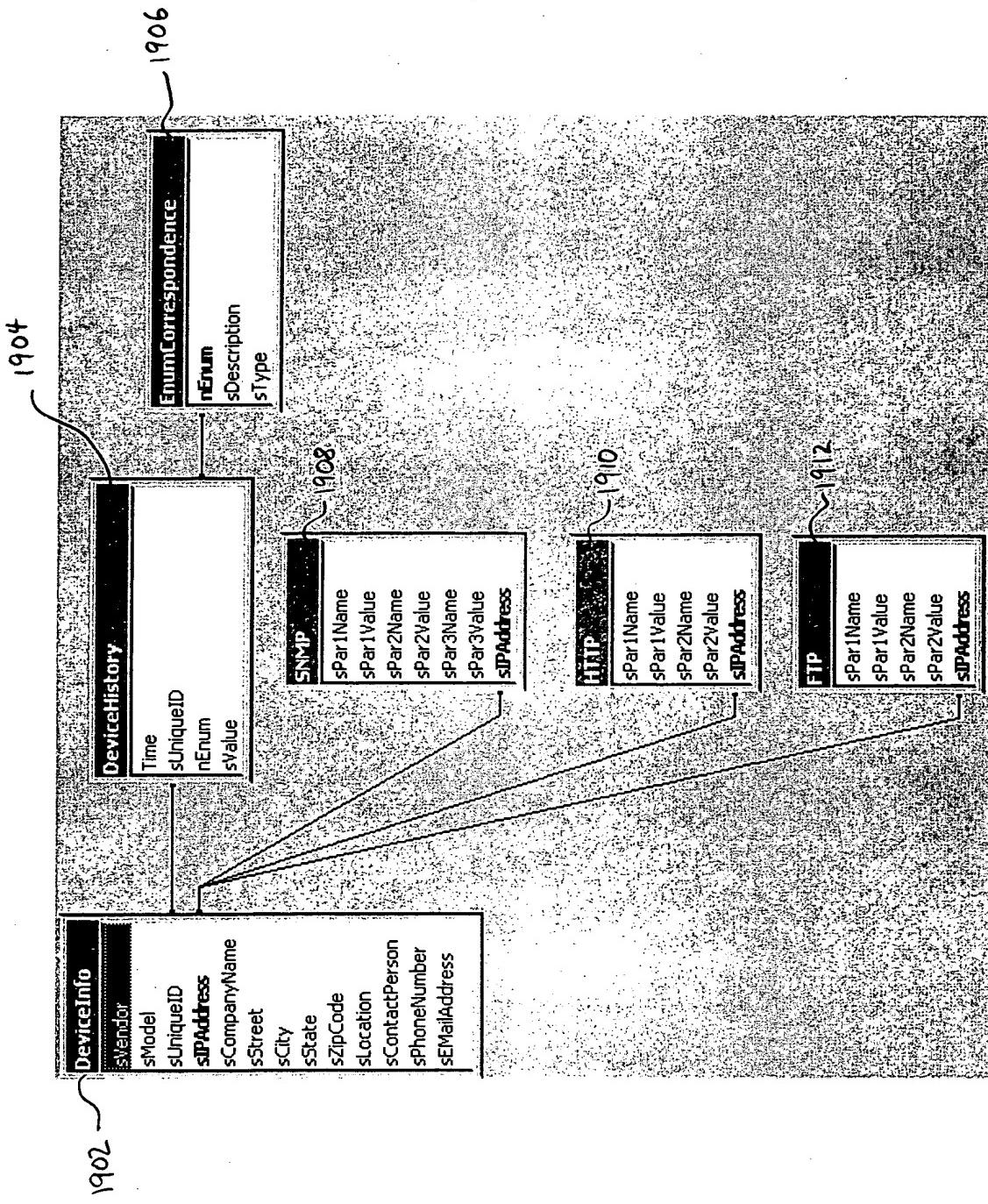


FIG. 19

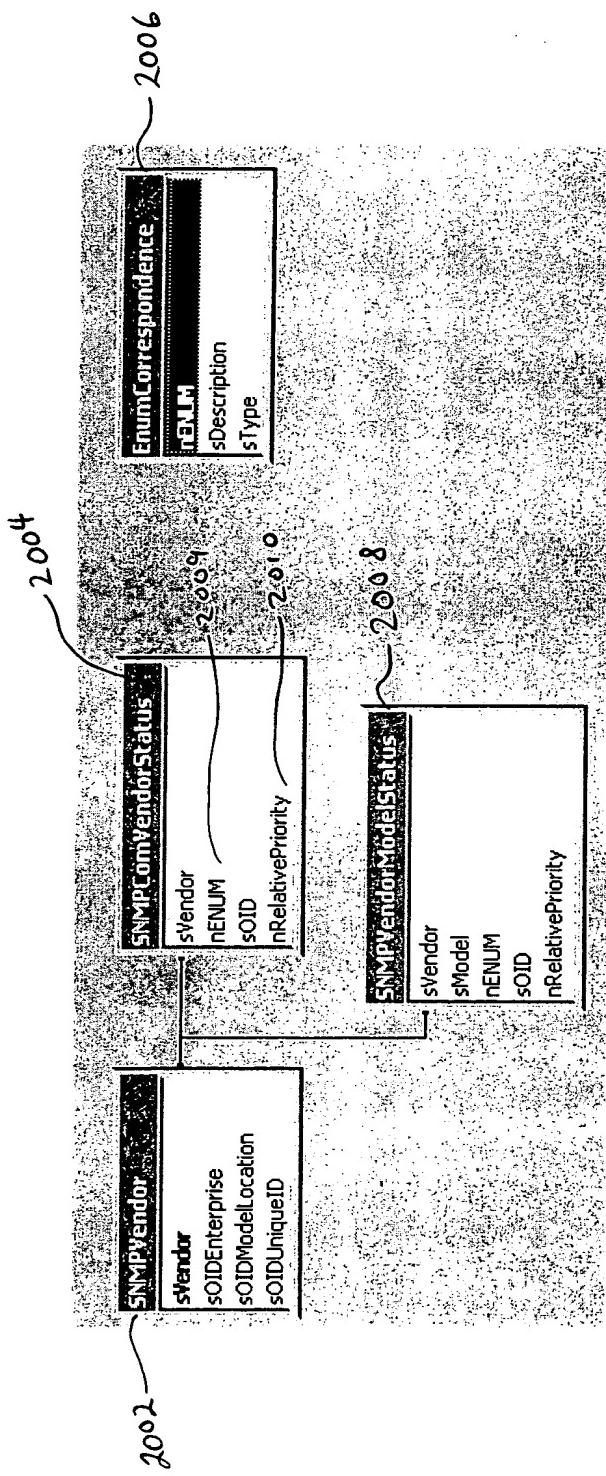


FIG. 20

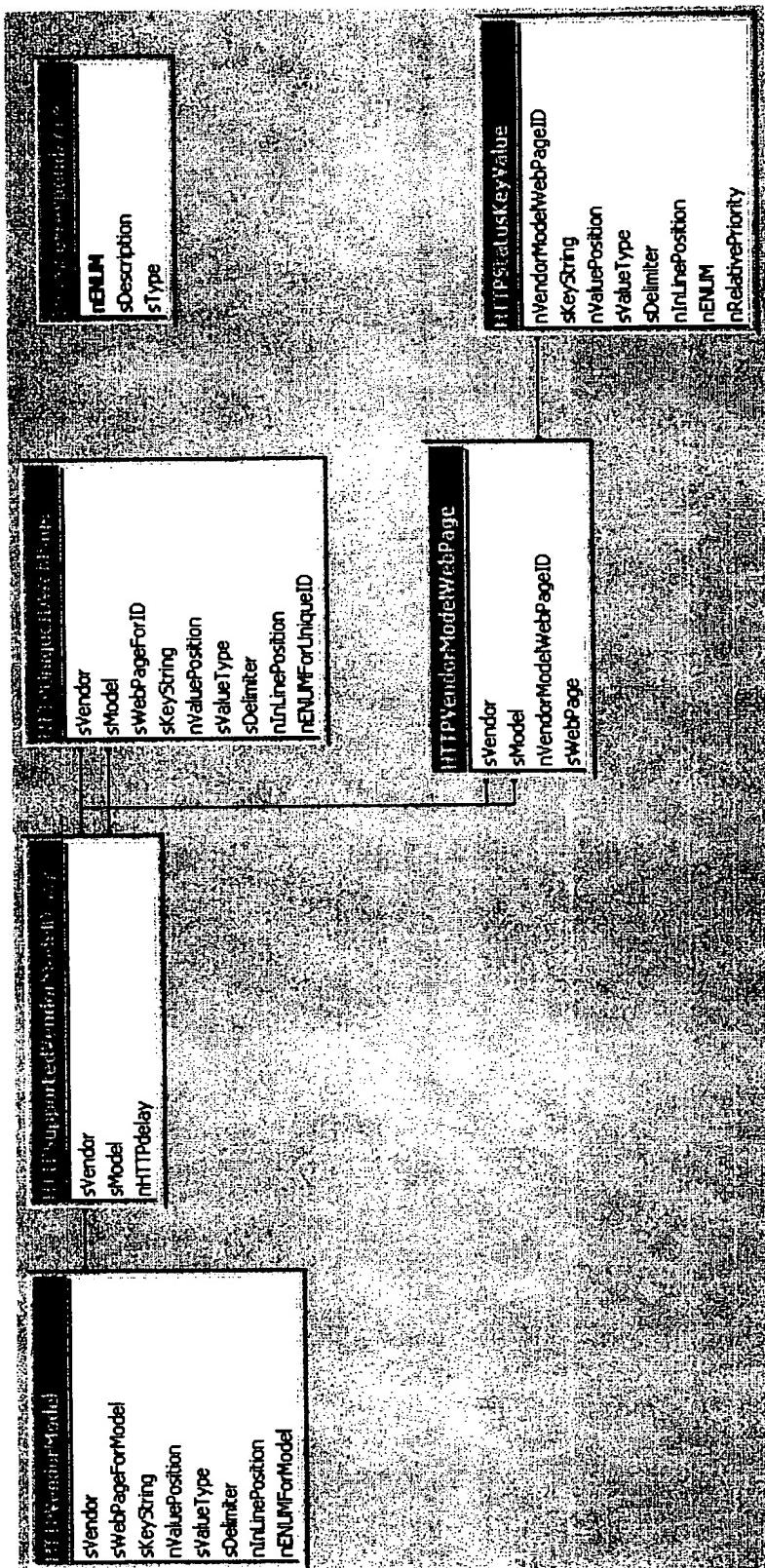


FIG. 21

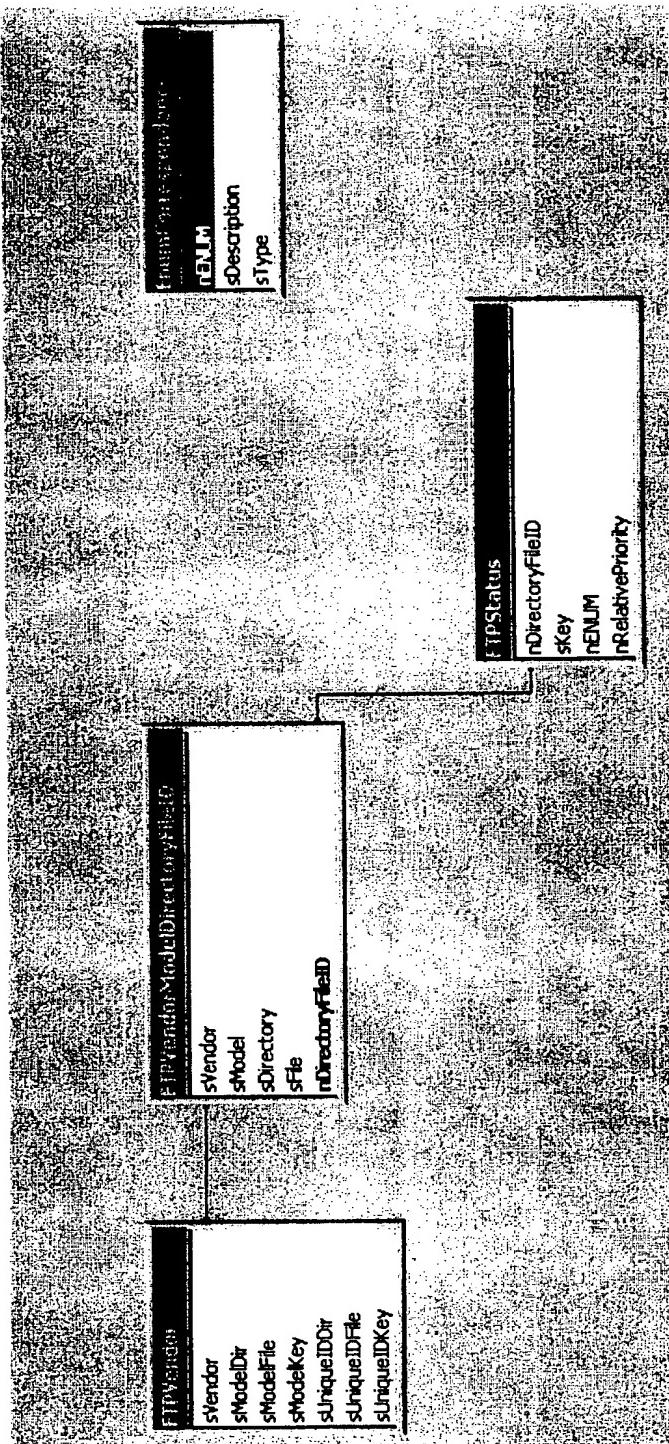


FIG. 22

2300

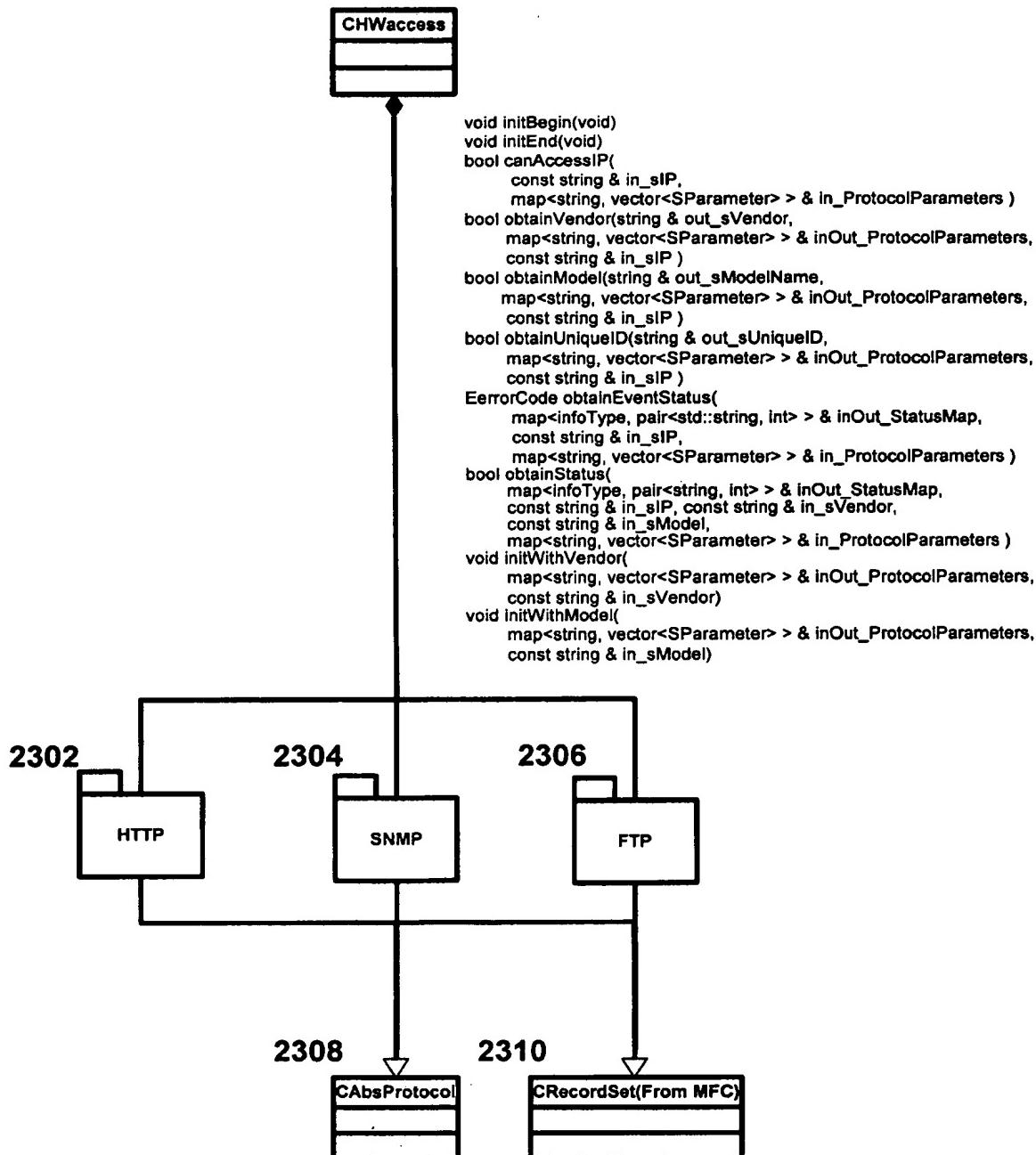
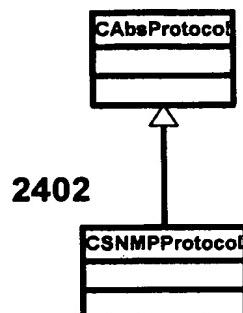


FIG. 23

2400

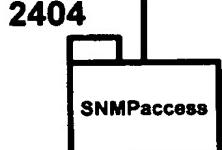


```

bool initiateSNMP(const string & in_sIP,
                   vector<SParameter> & in_Parameter)
bool obtainValueFromGetRequest(
    string & out_sValue, const string & in_sOID)
bool obtainValueFromGetNextRequest(
    string & out_sValue, const string & in_sOID)
bool obtainValueFromGetNextRequest(
    string & out_sValue)
bool obtainData(map<InfoType, pair<string, int> > & inOut_Data,
               vector<pair<SOIDInfoType, int> > & in_SOIDInfoTypeVector)
bool closeSNMP()
    
```

```

void init()
bool obtainSNMPSupportVendorModel(string & out_sVendor,
                                   string & out_sModel)
bool obtainSNMPVendor(string & out_sVendor, string &
out_sOIDVendor,
                      string & out_sOIDModel, string & out_sOIDUniqueId)
bool obtainSNMPVendorStatus(
    vector<pair<SOIDInfoType, int> > & out_OIDInfoTypeWeightVector,
    const string & in_sVendor)
bool obtainSNMPModelStatus(
    vector<pair<SOIDInfoType, int> > & out_OIDInfoTypeWeightVector,
    const string & in_sVendor, const string & in_sModel)
    
```



2406



2408

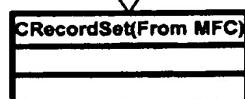


FIG. 24

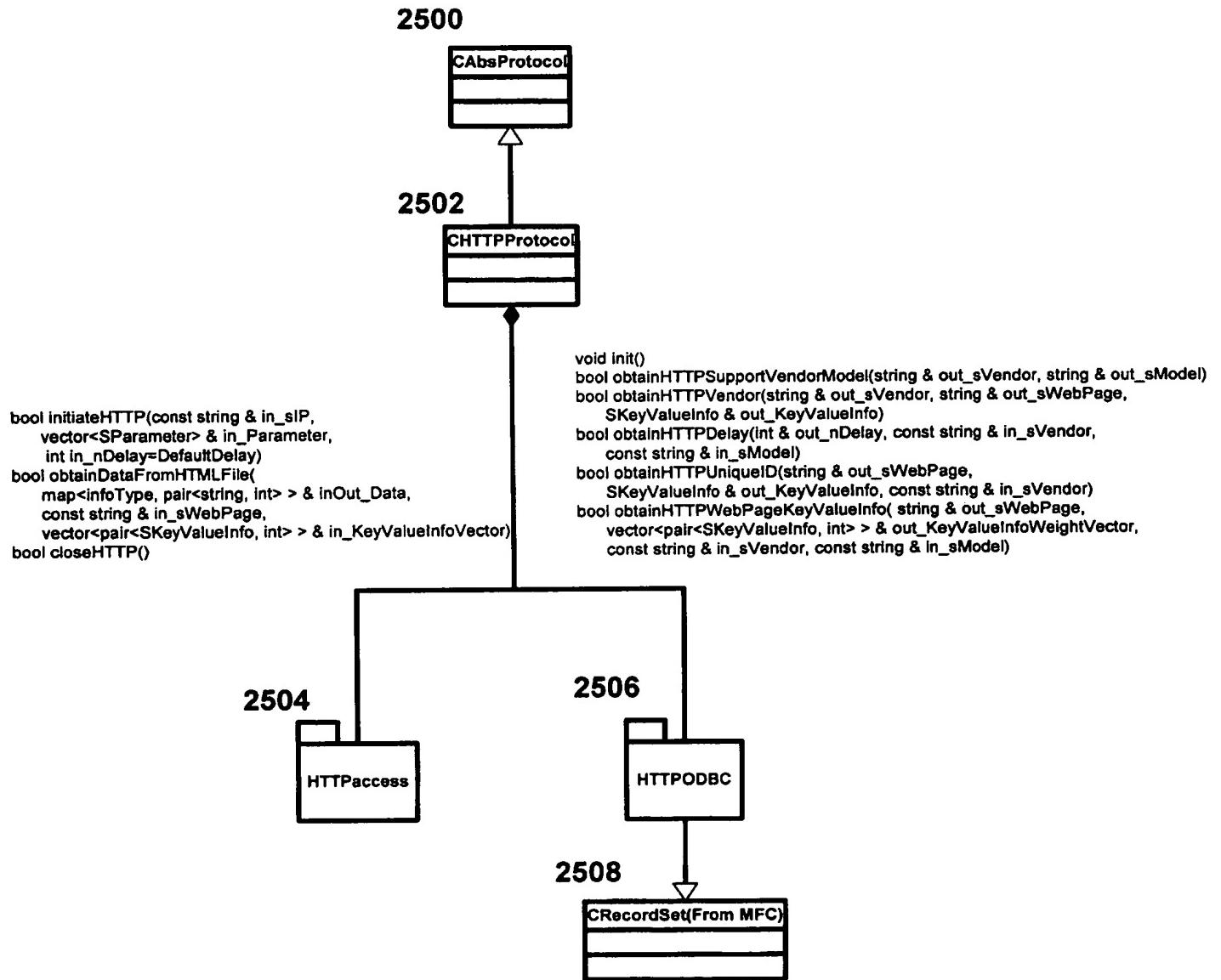


FIG. 25

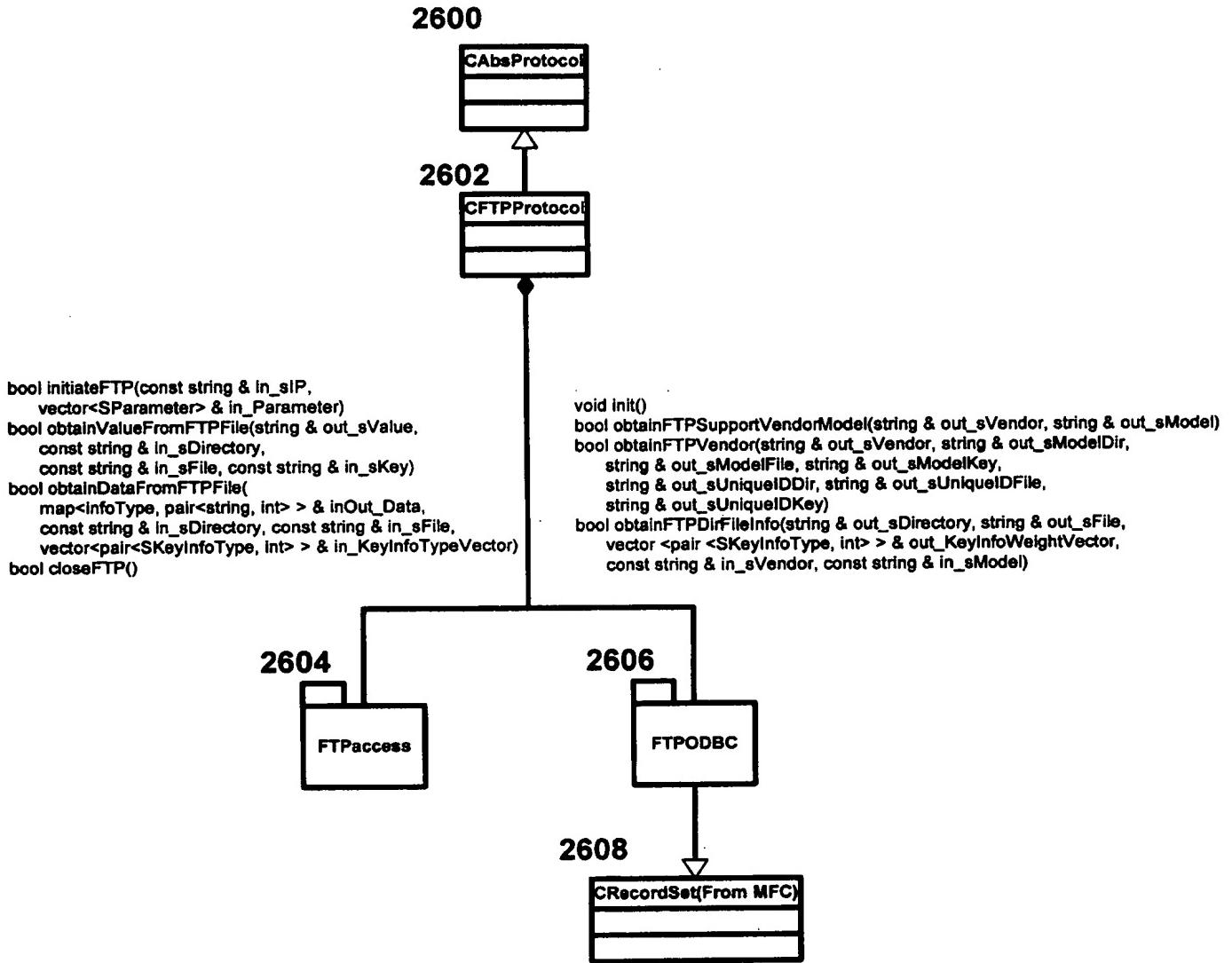


FIG. 26

Vector of CAbsProtocol*

500

CAbsProtocol* 502	CAbsProtocol* 504	CAbsProtocol* 506	...
------------------------------------	------------------------------------	------------------------------------	-----

FIG. 27A

std::map<std::string, std::map<std::string, std::vector<std::pair<SOIDinfoType, int> > >
510

string 512	map 514	
	string 516	vector 518
Ricoh	GENERIC	<(SOIDinfoType1, int1), (SOIDinfoType2, int2)>
	Aficio 1224C	<(SOIDinfoType3, int3)>
Xerox	GENERIC	<(SOIDinfoType4, int4), (SOIDinfoType5, int5)>
	DocuPrint NC60	<(SOIDinfoType5, int5)>

FIG. 27B

std::map<std::string, std::map<std::string, std::vector<SWebPageInfo>> >
520

string 522	map 524		
	string 526		vector 528
Ricoh	Aficio X	Vector of	<pre>struct SWebPageInfo { std::string m_sWebPage; std::vector<std::pair<SKeyValueInfo, int>> m_KeyValueInfoVector; };</pre>
Xerox	N4025		

FIG. 27C

std::map<std::string, std::map<std::string, std::vector<SDirFileInfo>> >
530

string 532	map 534		
	string 536		vector 538
Ricoh	Aficio	vector of	<pre>struct SDirFileInfo { std::string m_sDirectory; std::string m_sFile; std::vector<std::pair<SKeyInfoType, int>> m_KeyInfoTypeVector; };</pre>
Xerox	N4025		

FIG. 27D

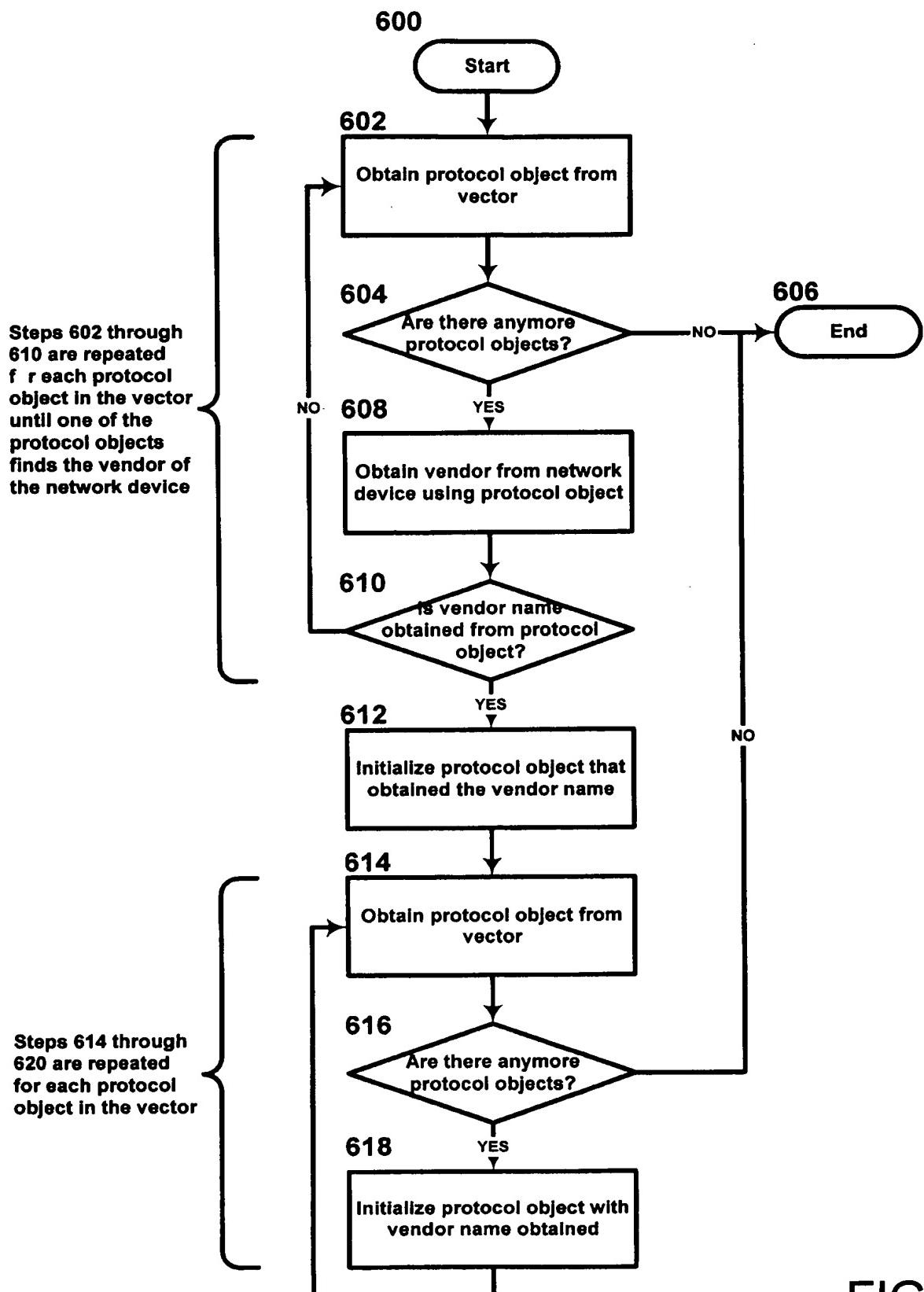


FIG. 28

SNMP Protocol
700

pair<SOIDInfoType, int> 702	pair<SOIDInfoType, int> 704	...
--	--	-----

FIG. 29A

706

```
struct SOIDInfoType {
    InfoType m_InfoType;
    std::string m_sOID;
    SOIDInfoType();
    ~SOLIDInfoType();
    void clear();
};
```

HTTP Protocol
708

pair<SKeyValueInfo, int> 710	pair<SKeyValueInfo, int> 712	...
---	---	-----

FIG. 29B

714

```
struct SKeyValueInfo {
    InfoType m_InfoType;
    std::string m_sKey;
    int m_nPosition;
    std::string m_sType;
    std::string m_sDelimiter;
    int m_nInLinePosition;
    SKeyValueInfo();
    ~SKeyValueInfo();
    void clear();
};
```

FTP Protocol
716

pair<SKeyInfoType, int> 718	pair<SKeyInfoType, int> 720	...
--	--	-----

FIG. 29C

722

```
struct SKeyInfoType {
    InfoType m_InfoType;
    std::string m_sKey;
    SKeyInfoType();
    ~SKeyInfoType();
    void clear();
};
```

Status Information Map
724

726

728

InfoType	pair<string, int>
infoType	pair<string, int>
InfoType	pair<string, int>
...	...

FIG. 29D

`std::map<std::string, std::map<std::string, std::vector<SDirFileInfo> > >`
800

Ricoh	Aficio 120	<SDirFileInfo1, SDirFileInfo2, SDirFileInfo3>

802

/pub
status.txt
<(SKeyInfoType1, 1000), (SKeyInfoType2, 5000), (SKeyInfoType3, 10000), (SKeyInfoType4, 7500), (SKeyInfoType5, 625)>

SKeyInfoType1 corresponds to the infoType 600, SKeyInfoType2 corresponds to the InfoType 610, SKeyInfoType3 corresponds to the InfoType 620, SKeyInfoType4 corresponds to the InfoType 700, and SKeyInfoType5 corresponds to the InfoType 710

Status Information Map

804

600	("Low Paper", 500)
610	("24321", 10000)
700	("OK", 2500)

FTP Protocol

806

(SKeyInfoType1, 1000)	(SKeyInfoType3, 10000)	(SKeyInfoType4, 7500)	(SKeyInfoType5, 625)
-----------------------	------------------------	-----------------------	----------------------

FIG. 30

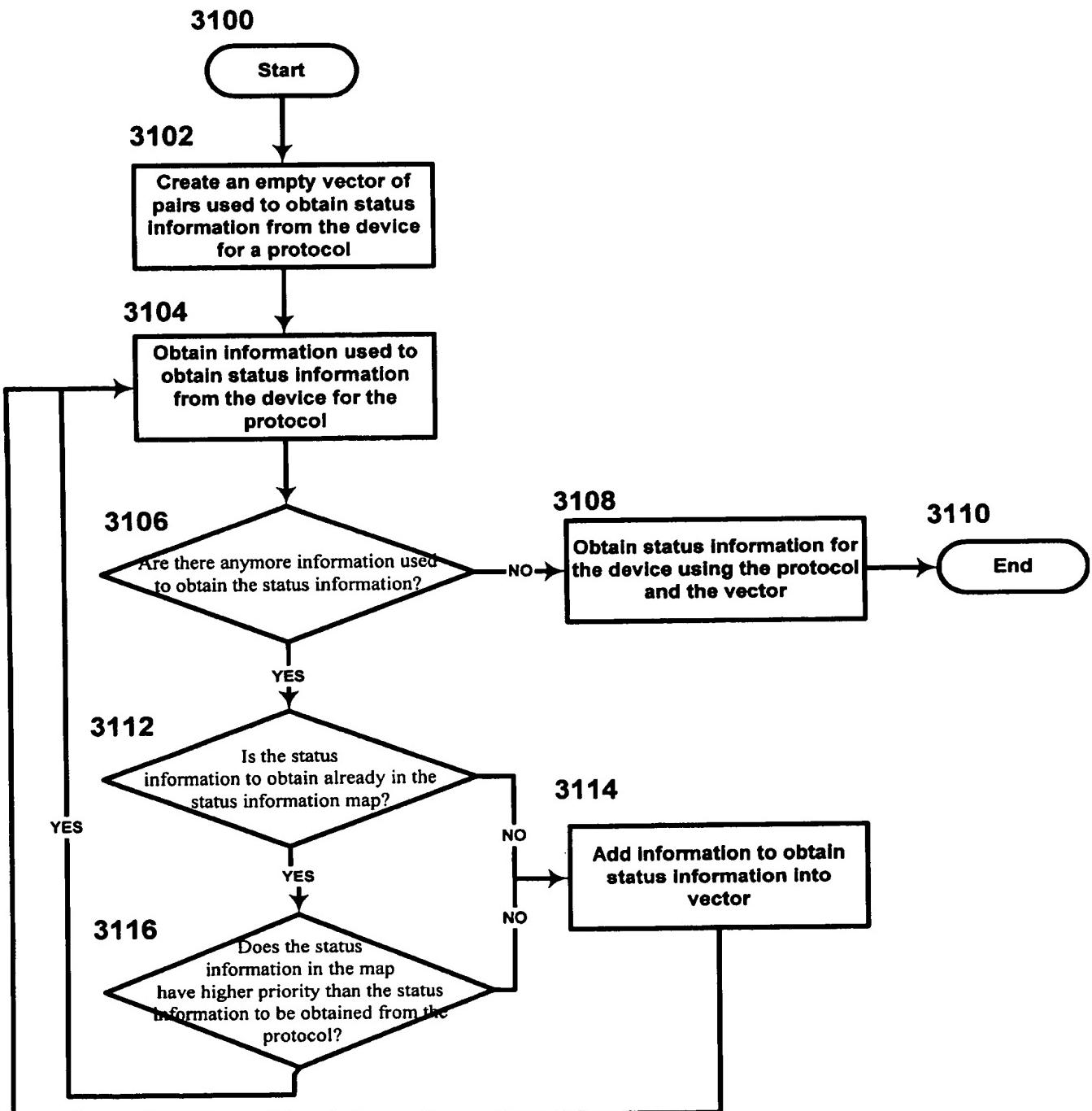


FIG. 31A

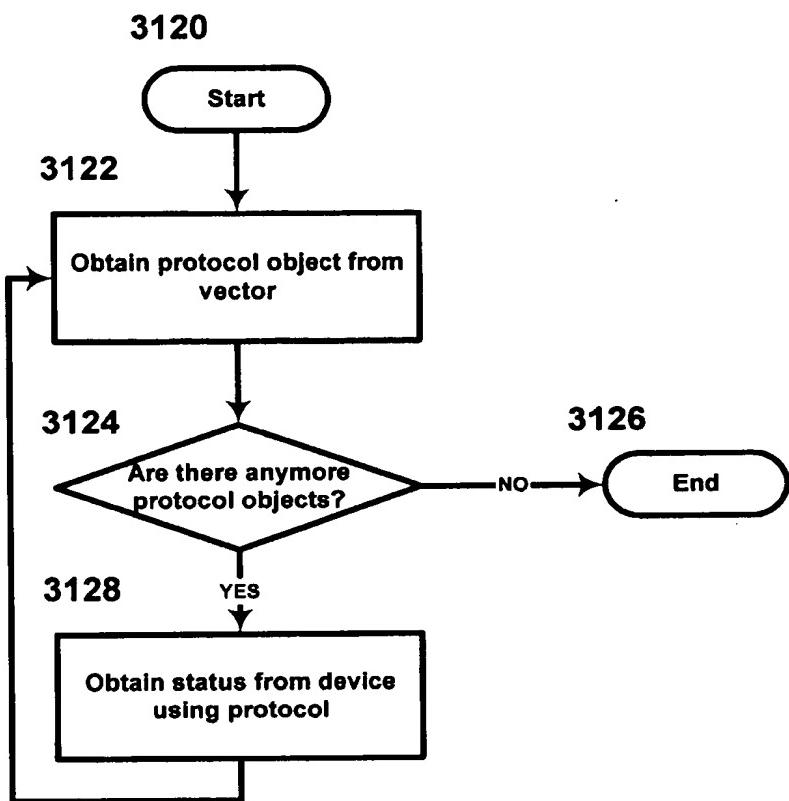


FIG. 31B

Vendor Model Support Map
3200

3202	3204
string	int
...	...

FIG. 32A

Sample Vendor Model Support Map
3206

3208	3210
Xerox%%%%%NC60	1
Xerox%%%%%N4025	1
HP%%%%%LaserJet 9000	1
HP%%%%%LaserJet 4550	1
...	...

FIG. 32B

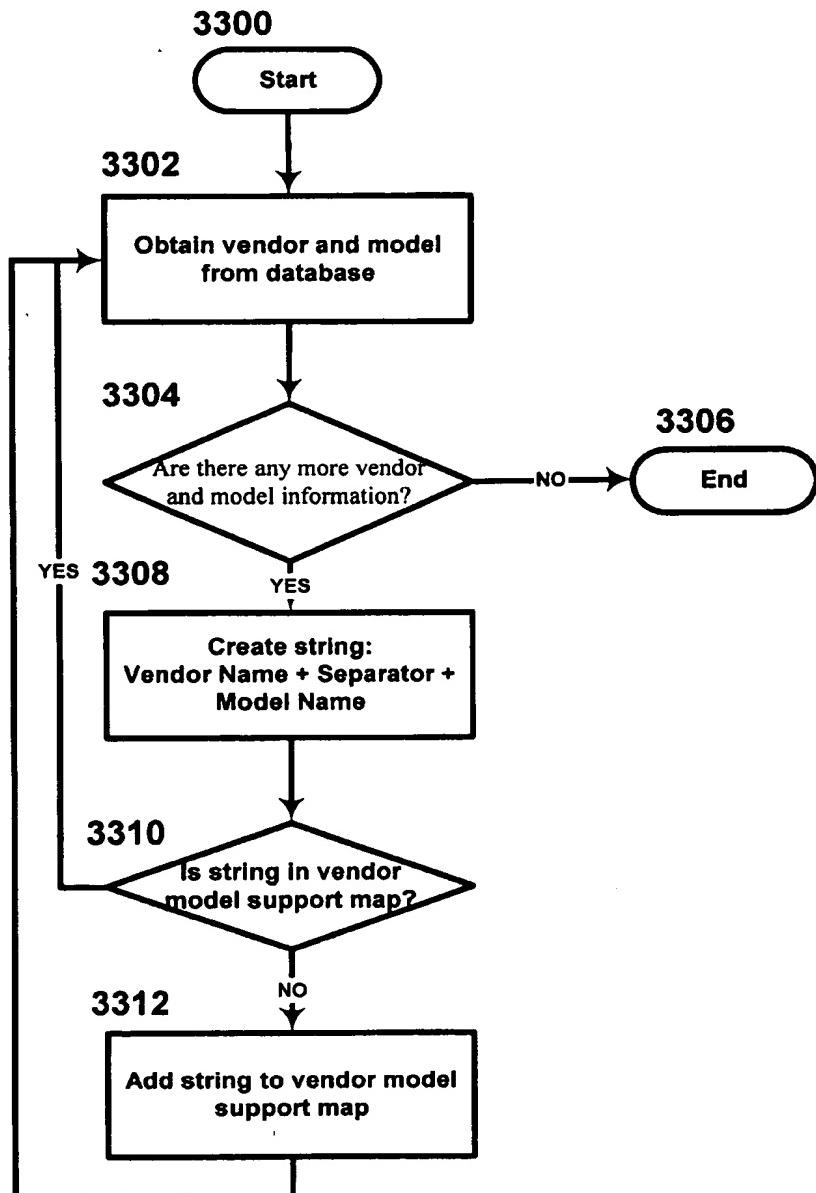


FIG. 33

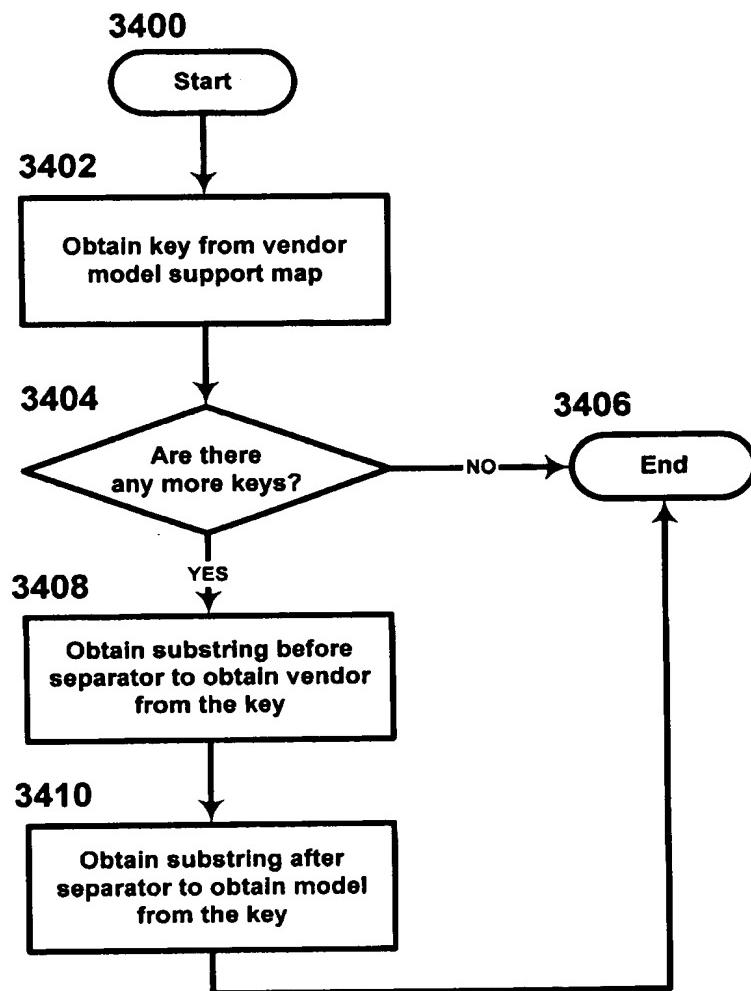


FIG. 34

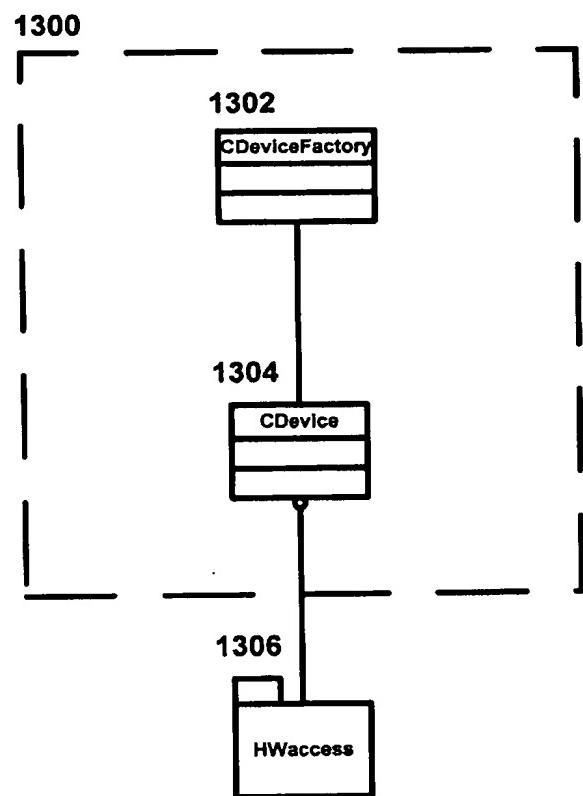


FIG. 35

Protocol Parameter Map
1400

1402

1404

1406

string	vector<SParameter>
...	...

```
struct SParameter {
    std::string m_sParName;
    std::string m_sParValue;
    SParameter ();
    ~SParameter ();
    void clear();
};
```

FIG. 36A

1410

SNMP	<(COMMUNITY, private)>
FTP	<(USERNAME, abc), (PASSWORD, xyz)>

FIG. 36B

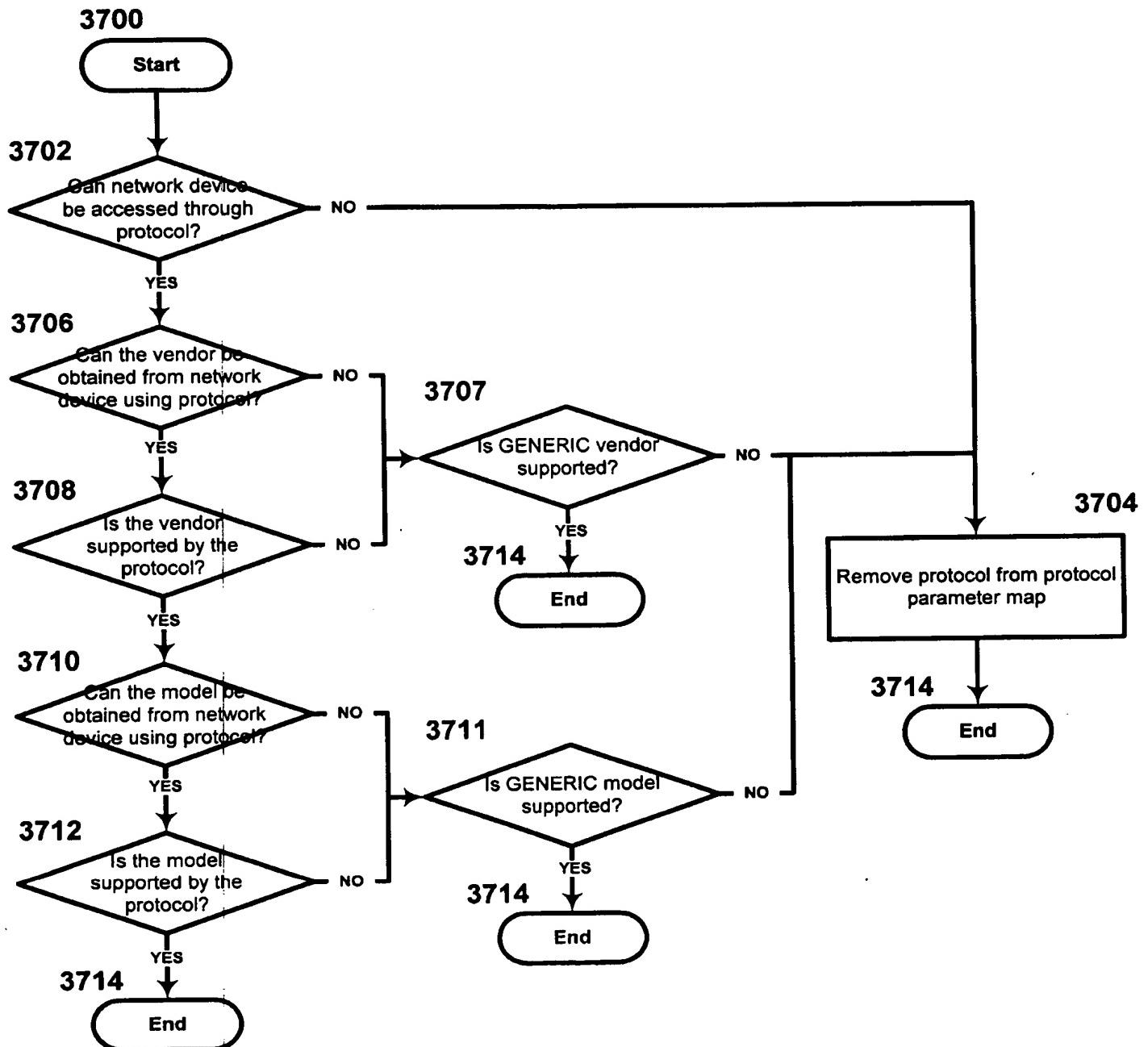


FIG. 37